A Flexible and Dynamic Mission Planning Architecture For UAV Swarm Coordination

Carlos Sampedro[1], Hriday Bavle[1], Jose Luis Sanchez-Lopez[1], Ramon A. Suárez Fernández[1], Alejandro Rodríguez-Ramos[1], Martin Molina[2], and Pascual Campoy[1]

Abstract— **In this paper a scalable and flexible Architecture for real-time mission planning and dynamic agent-to-task assignment for a swarm of Unmanned Aerial Vehicles (UAV) is presented. The proposed mission planning architecture consists of a Global Mission Planner (GMP) which is responsible of assigning and monitoring different high-level missions through an Agent Mission Planner (AMP), which is in charge of providing and monitoring each task of the mission to each UAV in the swarm. The objective of the proposed architecture is to carry out high-level missions such as autonomous multi-agent exploration, automatic target detection and recognition, search and rescue, and other different missions with the ability of dynamically re-adapt the mission in real-time. The proposed architecture has been evaluated in simulation and real indoor flights demonstrating its robustness in different scenarios and its flexibility for real-time mission re-planning and dynamic agent-to-task assignment.**

I. INTRODUCTION

Advances in sensor, computation and communication technologies have made the Unmanned Aerial Vehicles (UAVs) indispensable in areas where manned vehicles are too risky or in hazardous environments where the presence of human operators is limited, such as exploring areas with radioactivity risks [1], inspecting the interior of an entire building hit by an earthquake [2], image data acquisition in disaster areas [3], etc.

In the recent years, and due to its low cost and flexibility, swarm of UAVs have been increasingly investigated [4], [5], [6], [7], [8], [9], [10], [11], and utilized for different type of missions like search and rescue [12], automatic target detection and recognition [6], [13], [14], hunting a target employing multiple UAVs [15], and other high-level missions that require the coordination between several UAVs to perform the tasks in a faster and more efficient way. Several biologically-inspired approaches and strategies have been proposed for controlling the swarm behavior, such as strategies based on pheromones [6], [4], or using evolutionary algorithms [5], [7], [4].

The interaction between a swarm of UAVs can be seen as a multiagent system where several UAVs perform the tasks with communication and coordination amongst themselves. Thus, research efforts must aim towards the development of versatile and flexible architectures for optimal coordination of a swarm of robotic agents. Boskovic et al. [7] proposed a six-layered hierarchical architecture, where the problems of real-time mission re-planning, and dynamic agent-to-task assignment were addressed by combining mission planning using evolutionary algorithms, hybrid automata-based task execution and biologically-inspired emergent swarm behaviors.

A decisional architecture was proposed by Gancet et al. [16], in which several deliberative components were designed for multi-UAV system applications. In this work, the difference between operational and decisional autonomy for designing a multi-UAV architecture was exposed. In the case of a decisional autonomy, a possible division in several levels was defined, where the higher levels correspond to higher decisional capabilities, such as delegating planning activities to the UAV, mission and task refinements, task re-allocation, etc. Simulation results showed that the proposed decisional architecture was able to deal with high-level decisional requirements.

Gaudiano et al. [4] proposed a strategy using a Genetic Algorithm to evolve swarm control parameters, such as the transition probabilities of a UAV across different modes, the pheromone decay rate, and the pheromone attraction parameters of a UAV for search and supression of enemy air defense missions. Similarly, In Dasgupta [6] the swarming mechanism for automatic target recognition was based on the communication mechanism of insects using pheromones as a positive reinforcement for finding the trail to the target. The results presented were validated only in simulation scenarios (AEDGE simulation platform).

In the same line of scope, Lamont et al. [5] proposed a mission planning system for swarm of UAVs. This system consisted of a combination of several modules for mission planning, path planning using multi-objective evolutionary algorithms for terrain following missions, a genetic vehicle routing algorithm for vehicle-to-target assignment, and a swarm behavior module that tries to maintain some tendencies for the agents of the swarm such as remain together, maintain safe distance from one another, etc. Again, the proposed approach was tested only in a simulation environment (SPEEDES simulation framework), showing a limited dynamic re-planning and scalability.

These biologically-inspired strategies, such as genetic algorithms, present a feasible solution for autonomous mission planning due to their capacity of avoiding computational complexity, obtaining a solution very close to the optimal

*The authors would like to thank the Spanish Ministry of Science MICYT for project funding (DPI2014-60139-R) and CSIC for a JAE-Predoctoral scholarship.

[1]All authors are with the Computer Vision Group, Center for Automation and Robotics (UPM-CSIC), Calle José Gutiérrez Abascal 2, Madrid (Spain). carlos.sampedro@upm.es

[2]Department of Artificial Intelligence, Technical University of Madrid (UPM) (Spain).

one. However, convergence of this kind of algorithms can be very slow due to its unguided mutation, making them less suitable for real-time purposes.

Waharte and Trigoni [12] proposed three strategies for search and rescue operations, which were studied and evaluated based on the time to find the target. In the greedy heuristics approach, each UAV moves to the neighboring cell based on the highest belief confidence. In the potential-based approach attractive and repulsive potentials were created for navigating through the obstacles present in the scenario, searching for the target. The Partially Observable Markov Decision Process was studied for creating different observation models and a set of actions for each set of UAVs. The results obtained were presented only in simulation cases of study.

In Wei et al. [9] the problem of dynamic mission planning was investigated. A centralized-distributed framework based on a central controller was proposed, being responsible for the mission assignment to each UAV in the swarm. In order to monitor the mission, the central controller periodically sends status inquiries to all UAVs. Unlike the previous framework, we propose a centralized, dynamically distributed and flexible mission planning architecture based on a Global Mission Planer (GMP) and an Agent Mission Planner (AMP). In the proposed architecture, the AMP inquires the GMP only when needed, e.g the searched target in the mission is found and a new mission has to be assigned.

In the related works, most of the approaches presented were only tested in simulation environments with several assumptions made for particular cases of study. In this paper a dynamic and flexible architecture is proposed with the aim of being fully-operative and adaptable to real time constraints. The proposed architecture has been tested and evaluated in both simulation as well as in real indoor flights with different swarm configurations, addressing the problems of scalability and flexibility. In addition, several high-level missions, such as Target Detection and Exploration, have been planned for testing the response of the proposed mission planning architecture to heterogeneous and dynamic high-level missions.

The rest of the paper is structured as follows: Section 2 presents the problem statement and the motivation. In Section 3 the proposed architecture is described. Section 4 reports the experiments performed in simulated and real scenarios, with their respective results. And section 5 concludes the paper, as well as points towards future research directions.

II. PROBLEM STATEMENT AND MOTIVATION

High-level missions, such as automatic target detection and inspection, or autonomous exploration of an area using a UAV can be very challenging. These challenges increase significantly in indoor environments where there is no availability of GPS data for global localization. To efficiently perform such high-level missions with a high level of autonomy, a robust and flexible architecture is needed. This architecture has to be composed by several core modules like a localization module for estimating the pose of the UAV in the world, a mission planner to assign and monitor the individual tasks of the mission, a trajectory planner to assign trajectories with obstacle avoidance, a trajectory controller to move the UAV along the desired trajectories given by the trajectory planner, and several modules for error handling and monitoring, communication and supervision.

The need of such kind of architectures is emphasized for complex missions that have to be performed in the industry environment. Missions, such as power lines inspection, bridges inspection, etc., require the use of a complex and coordinated system for optimizing the resources while completing the mission. Performing these tasks with only one UAV would require, in most of the cases, several attempts of inspection due to the endurance of the actual batteries. In these kind of scenarios the advantages of using a swarm of UAVs can lead to an efficient management of the resources with the consequent saving of money for the companies. In addition, in several kind of missions, where the time of accomplishment is a critical constraint (e.g. search and rescue), a swarm of UAVs can be very suitable to efficiently perform the required mission.

In contrast, a swarm of UAVs is a complex system where the different agents need to be efficiently coordinated. For this purpose, the effort has to be focused on designing a scalable and flexible architecture, capable of managing all the modules of such a complex system. Our previous framework [11] has all the capabilities for this management, but does not have a global coordinator capable of managing a multi-robot swarm of UAVs through the execution of a global mission, that can be dynamically re-adapted at any moment if some event occurs (e.g. the searched target has been found).

This paper is an effort in this direction. Thus, based on all the functionalities of our current architecture, two new modules have been designed for having a global coordinator called Global Mission Planner (GMP), which is in charge of dynamically and efficiently distributing the missions through all the agents in the swarm, while monitoring the swarm behavior, and a local coordinator called Agent Mission Planner (AMP), which is in charge of executing these tasks with an asynchronous communication with the GMP.

III. SYSTEM ARCHITECTURE

In this section, the proposed architecture is described in detail. In section III-A, a general overview of the global architecture is presented. Sections III-B and III-C explain the functionality of the Mission Planning architecture of the proposed approach.

A. *GENERAL ARCHITECTURE*

With the objective of having a fully autonomous flight-proven swarm of multi-UAV agents, Aerostack[1] has been employed. Aerostack has a fully distributed swarm design with no-coordination between robotic agents. A full description of Aerostack architecture and its components is out of

[1]Aerostack webpage: `http://www.aerostack.org/` and Aerostack Github repository: `https://github.com/Vision4UAV/Aerostack`

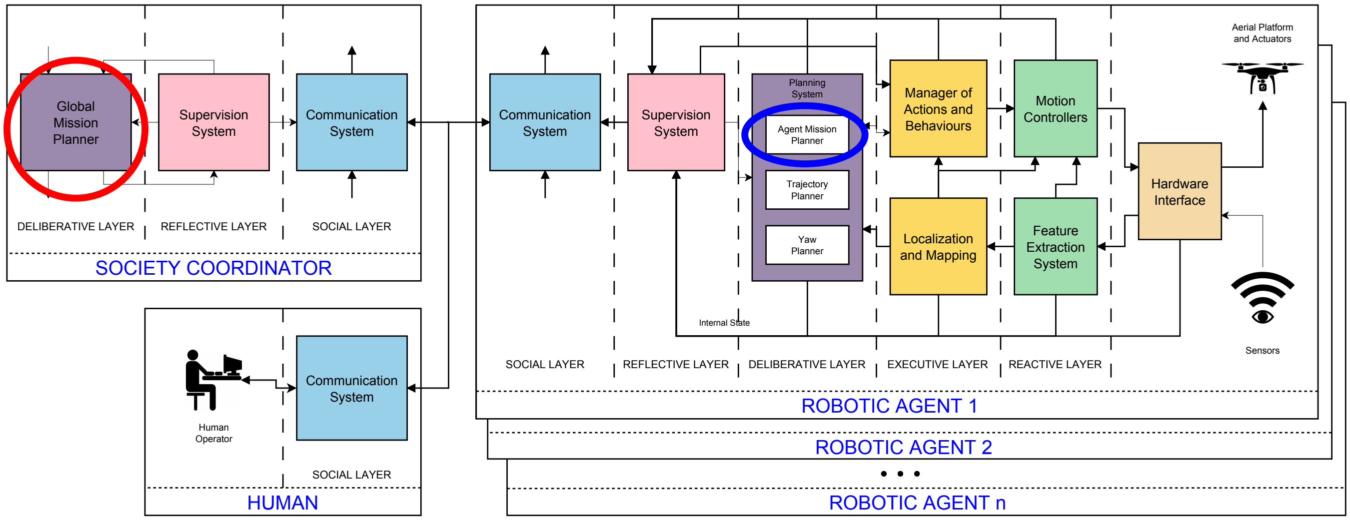


Fig. 1: Aerostack architecture diagram with Swarm Mission Coordinator. The Society Coordinator includes the GMP (surrounded with a red circle) that plans high-level missions and coordinates all the robotic agents. Every Robotic Agent is able to execute a fully autonomous mission through its AMP (surrounded with a blue circle) and using all its components (described in [17]). The Human is able to request high-level missions to the system and monitor their progress.

the scope of this paper and can be reviewed in [11] and [17]. Using Aerostack, every robotic agent is able to perform a fully autonomous mission thanks to a set of interrelated and robust components. Aerostack has a set of available actions (e.g. take-off, land, hover, move to point, and turn in yaw) and behaviors (e.g. recognize visual markers) together with its performance and state, that can be directly used ensuring a fully autonomous operation.

To robustly implement the proposed GMP, a new meta-component has been included in Aerostack, called Society Coordinator, that fully communicates with the rest of the Robotic Agents and Human Operators (see Figure 1).

The Society Coordinator follows Aerostack architecture design and it has three main components: The Communication System allows and ensures the communication between this module and the rest of the robotic agents in the swarm as well as the human operator. The Supervision System has the cognitive and self-awareness of the whole Society Coordinator, ensuring its correct operation. The GMP, described in section III-B, is in charge of planning the high-level missions requested by the user, and coordinating the whole society.

In order to ensure the compatibility with the GMP, a new AMP has been also developed and added to Aerostack that plans and monitors, task by task, the missions assigned to every specific agent in the swarm (see section III-C).

B. GLOBAL MISSION PLANNER

In this section the general intelligence of the GMP is presented. This module of the architecture has been designed for being able to dynamically manage high-level missions, such as Search and Rescue, Target Detection, Surveillance, etc, providing a higher level of intelligence in architecture terms, while being much more easy-to-use for a human operator. Thus, it receives as input from a configuration file, the high-level mission (e.g. Find Target, Explore, etc) to be accomplished, and the dimensions of the area in which the mission has to be performed (referred by us as Mission Zone). The Communication System provides the number of agents in the swarm, which report their available state to the GMP. Once the global mission has been established (e.g. Find a Target), the GMP proceeds according to the following criteria:

- Sample the Mission Zone. The objective of this step is to divide the area in which the mission has to be performed in several regions. For this purpose a sampling algorithm based on k-means clustering has been utilized. The procedure consists of randomly distributing points through the Mission Zone and perform k-means clustering until convergence with the K clusters specified in the configuration file of the mission.
  This is done in two steps, the assignment step in which each sample in the Mission Zone is assigned to the closest cluster (Eq. 1), and an update step (Eq. 2), in which each centroid of each cluster is recalculated based on the actual members of the cluster. The algorithm converges when the assignments no longer change, satisfying (Eq. 3).

$$C_i^{(t)} = \{x_p : \|x_p - \mu_i^{(t)}\|^2 \leq \|x_p - \mu_j^{(t)}\|^2 \\ \forall j, 1 \leq j \leq K\} \tag{1}$$

$$\mu_i^{(t+1)} = \frac{1}{|C_i^{(t)}|} \sum_{x_j \in C_i^{(t)}} x_j \tag{2}$$

$$argmin_c \sum_{i=1}^{K} \sum_{x_j \in C_i} \|x_j - \mu_i\|^2 \tag{3}$$

Where $C_i^{(t)}$ denotes the i_{th} centroid of the cluster at iteration t, x_p is a point belonging to the Mission Zone, $\mu_i^{(t)}$ is the mean of i_{th} cluster at iteration t, and K is the total number of clusters to be computed.
As a result of this step, a Voronoi diagram is obtained (see Figures 4a, 4b, 4c), in which the centroids of each cluster will be used by the GMP to build the mission points.

- Generate the mission points. In this step the GMP calculates the points, in the euclidean space, that are going to be visited or explored by the agents in the swarm. The mission points correspond to the centroids of the clusters that have been calculated in the previous step.
- Distribute the mission points between the agents in the swarm. In this function resides most of the computation of the GMP. This function takes as inputs the mission points calculated in the previous step as well as the take-off point of each UAV in the swarm, and generates a list of mission points per UAV. The calculation criteria for this purpose is divided into 2 steps:
  1) Distribution of the mission points. Each mission point is assigned to the closest UAV in the Mission Zone, according to Eq. 4.

$$U_i = \{C_p : \|C_p - T_i\|^2 \leq \|C_p - T_j\|^2 \quad \forall j, 1 \leq j \leq N\} \tag{4}$$

  Where U_i denotes a set of mission points assigned to the i_{th} agent, C_p is the centroid of cluster p, T_i is the take-off point of the i_{th} agent, and N is the total number of UAV agents in the swarm.

  2) Arrangement of mission points based on the progressive euclidean distances along the path. The output points from previous step (step 1) are then sorted for obtaining a feasible path. For this purpose, the following criteria is applied:

for $i \in \{1, \ldots, N\}$ **do**
  for $j \in \{1, \ldots, U\}$ **do**
    for $z = j \in \{1, \ldots, U\}$ **do**
      $d_z = \|U_{ij} - U_{iz}\|^2$
    end for
    $m = min\{d\}$
    $swap\{U_{ij}, U_{im}\}$
  end for
end for

  Where d is a vector of distances between mission points, and m denotes the index of the minimum element in vector d.

- Generate tasks for each UAV in the swarm. Taking into account all the mission points assigned to each UAV in the swarm, and depending on the global high-level mission to be accomplished, the GMP generates different type of tasks, such as:
  – Take-off. This task consists of performing the transition from Landed state to Taking-off state, in which the UAV increments its z coordinate.
  – Hover. The task of Hover maintains the UAV at the current pose.
  – Move To Point. This kind of task implies the displacement of the UAV in any of the (x,y,z) coordinates.
  – Turn In Yaw. When this task is scheduled, the UAV turns only in the yaw angle without changing its 3D position.
  – Land. The accomplishment of this task implies that the UAV lands at the current position.

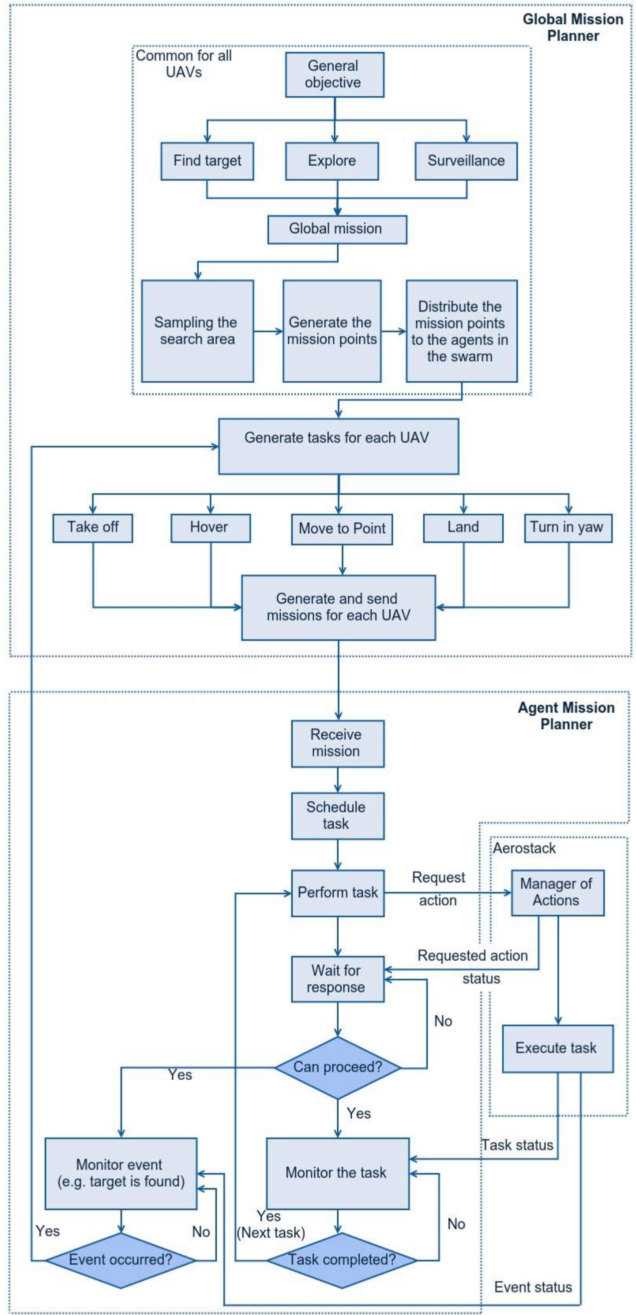


Fig. 2: Mission Planning communication architecture.

The GMP has some basic tasks (e.g. Take-off, Hover, Land) that are generated independently of the high-level mission to be performed. Other specific tasks (e.g. Move To Point, Turn in Yaw, etc.) are generated by the GMP depending on the high-level mission specified by the human operator in a configuration file.

- Generate the Mission to be performed by each UAV in the swarm. Once the tasks have been generated, the GMP is in charge of the arrangement of the tasks in the correct order for building a complete mission. As an illustrative example, giving 2 tasks of *MoveToPoint*, a complete mission for a UAV agent in the swarm could be: $\{$Take-off, Hover, MoveToPoint1, MoveToPoint2, Land$\}$.
- Send each Mission generated for each UAV in the swarm through the Agent Mission Planner.
- Mission re-planning. If some event occurs during the mission, the GMP has the ability of dynamically change the global mission in real-time and re-assign it to each UAV in the swarm.
  As an example, if the high-level mission of Find Target has to be performed, and if one agent in the swarm finds the target, a signal of mission accomplished is sent from the AMP to the GMP, which breaks the current mission and assigns a new one to each UAV in the swarm. In the experiments carried out in this work, and concretely for the Find Target mission, the re-planning of the mission consists of commanding the UAV that has found the object to inspect it and send the rest of UAVs in the swarm to the initial take-off point.
  The GMP has a default behavior as explained before, but some other behaviors (e.g. commanding several UAVs to inspect the target) can be easily integrated and specified by the human operator in the configuration file, providing high flexibility to the GMP.

C. *AGENT MISSION PLANNER*

The objective of the AMP is to plan and execute the mission given by the GMP in each UAV agent of the swarm. The functionality of the AMP proposed in this architecture is similar to the mission planner present in our previous architecture [11], the difference being that the AMP gets the mission dynamically from the GMP whereas in the previous architecture the user has to manually assign the mission for each UAV agent using an xml based language.

The AMP is a module with partial knowledge as compared to the GMP. It does not have any knowledge of what type of mission it is executing, like whether it is performing a Find Target mission or an Exploration mission. The AMP acts as a bridge between a high-level mission and the execution of the actions required for the accomplishment of such mission.

After receiving the complete mission from the GMP, the AMP starts scheduling the tasks of the received mission. After scheduling the tasks, the AMP starts performing the given tasks of the mission sequentially. During the execution of a task, the AMP translates it into an action (e.g., takeoff, move to a point, etc.), and requests for this specific action to the module in the Aerostack called Manager of Actions. The Manager of Actions in return translates the requested actions into specific commands for the motion controllers and activates the processes that are needed for the performance of the specific task (See Figures 1 and 2 for more details). The AMP then monitors the task completion before starting the subsequent task.

The AMP has the ability to simultaneously monitor an event while performing the mission (e.g. Find Target). When an event occurs out of the scope of the task-by-task mission execution (e.g. target has been found) the AMP instantly breaks the current mission, requesting the GMP for a new one, and dynamically re-adapts itself from the current mission being performed to the new received one.

IV. EXPERIMENTS AND RESULTS

This section begins describing the experimental set-up that has been conducted in order to evaluate and test the behavior of the proposed Mission Planning architecture in concordance with the general architecture of our framework presented in Section III-A. Subsequently, the experimental methodology is presented, in which we explain the experiments that have been carried out in order to test the different high-level missions commanded from the GMP, both in simulation as well as in real indoor flights (see Table I).

TABLE I: Experiment Set performed for real and simulated flights.

Mission	Map Size (m^2)	Type of flight	# UAVs	# Obs.
Find Target	9x10	Real & Sim.	2	6
Find Target	20x20	Simulated	4	11
Explore	20x30	Simulated	6	12

A. *Experimental Set-up*

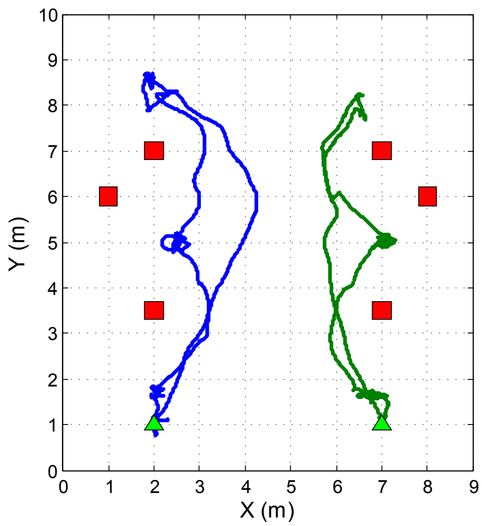
In this section, the experimental set-up that has been utilized for real and simulated flights is described in detail. In both cases, several computers have been used for processing all the software architecture running during flights. The software used in the proposed architecture is built in C++, under the standard C++11, using ROS (Robot Operating System) [18] as the communication framework between the different components in the architecture.

For localization and pose estimation purposes, square fiducial markers (also known as Augmented Reality Markers) named as ArUco [19] have been utilized. These markers provide a robust and real-time pose estimation, that is used in the proposed experiments for navigation inside the Mission Zone and for obstacle avoidance.

1) Real Flights: A real flight has been conducted in order to test and evaluate the proposed architecture. For this purpose, a flying area of $9 \times 10\ m^2$ inside a building has been prepared (Fig. 3).

Using the Mission Zone presented in Figure 3, several flights have been conducted using two UAVs. The UAV platform selected for real flight experiments is the ArDrone 2.0, which provides a low-cost and robust platform for testing purposes.

(a) Picture of the real flight scenario.



(b) Trajectories generated by the 2 UAV agents.

Fig. 3: Real flight experiment performed in a $9 \times 10m^2$ map.

Inside the Mission Zone two kind of objects are presented:

- Landmarks. These objects are arranged around the perimeter of the Mission Zone, and are labeled by 2 visual markers. The purpose of the landmarks is to help the UAV in the localization inside the map (Fig. 3a, grey objects).
- Obstacles. These objects correspond to cylindrical objects of size $20cm$ of radius, and can be arranged at any position in the map (Fig. 3a, brown objects). These obstacles are labeled by 4 visual markers.

Due to the limitation in space and number of UAVs, a swarm of only 2 UAV agents has been utilized in the real flights for a map of $9 \times 10\ m^2$.

2) Simulated Flights: As mentioned above, due to the limitation of resources in terms of number of UAVs available, only real flights using a swarm of 2 agents were performed. Consequently, and with the purpose of evaluating the capability and flexibility of the proposed architecture, more experiments have been conducted varying the size of the Mission Zone, number of obstacles present in the Mission Zone, number of UAV agents that conform the swarm, and the type of high-level mission that has to be conducted (see Table I).

In this case, the simulator present in AeroStack architecture has been utilized for simulating the autopilot and mid-level controller of the UAV, as well as the sensors and environment, using the 3D visualization tool for ROS called Rviz.

B. Experimental Methodology and Results

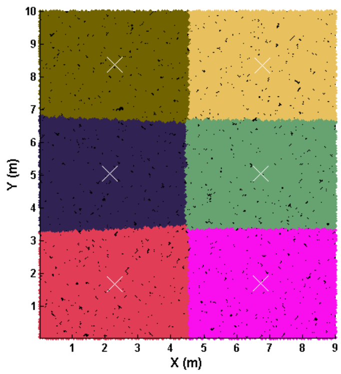
The experimental methodology proposed in this work has been thought for increasing the difficulty in each consecutive experiment, by means of modifying the size of the Mission Zone, the number of UAVs that conform the swarm, and the number of obstacles present in the Mission Zone (see Table I). Several specific high-level missions, such as Find Target or Explore, have been planned and simulated using the proposed experimental set-up of subsection IV-A. These high-level missions consists of:

- Find Target. In this type of mission the only input to the GMP is the high-level command of *FIND_TARGET*, given by the human operator. The objective of this mission is to find a predefined object that can be present in the Mission Zone. In the experiments conducted, an ArUco visual marker has been used as the object to be found. Thus, once one agent in the swarm detects the target, the Agent Mission Planner sends the event of *OBJECT_FOUND* to the GMP, which re-plans the mission in real-time according to the following procedure:
  - The agent that has found the object is commanded to perform a maneuver for showing the human operator that the target has been found (e.g. doing a flip), and is then commanded to inspect the target. Finally, once the inspection has been completed, the agent is sent to its initial take-off point, performing a landing maneuver when the initial point is reached.
  - The rest of agents in the swarm are commanded to go to their respective initial take-off points, performing a landing maneuver when the initial point is reached.

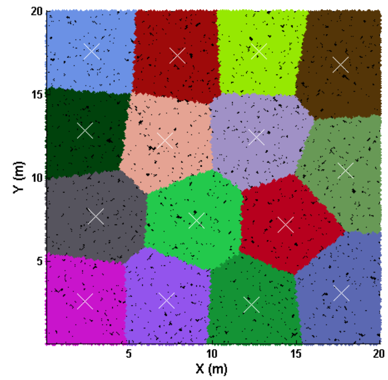
  The normal behavior of the agents of the swarm during the execution of the corresponding mission is as follows:
  - Move to point. The agents will move in trajectory control mode to the commanded 3D point that is established in the corresponding task.
  - Turn in yaw at different angular steps to complete a $360°$ turn, while maintaining the current position in (x, y, z). This behavior is performed for exploring the surroundings of the mission point reached by the UAV.

  The mission of *Find Target* will be finished if the target is found and all the agents in the swarm have returned to their respective initial take-off points, or when the object is not found and all the mission points have been reached.

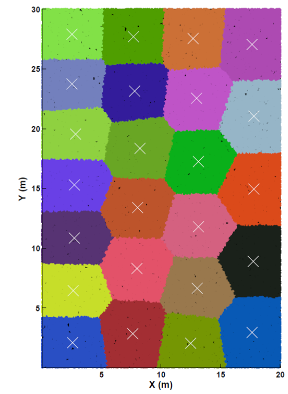
  Results of the experiments conducted for the *Find Target* mission are shown in Figures 3b, 4d, where the target is situated at point (2, 7, 1.3) and in Figure 4e,



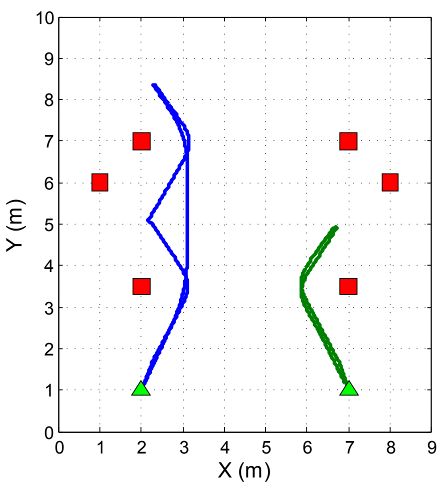
(a) Voronoi sampling regions in the $9 \times 10m^2$ map.



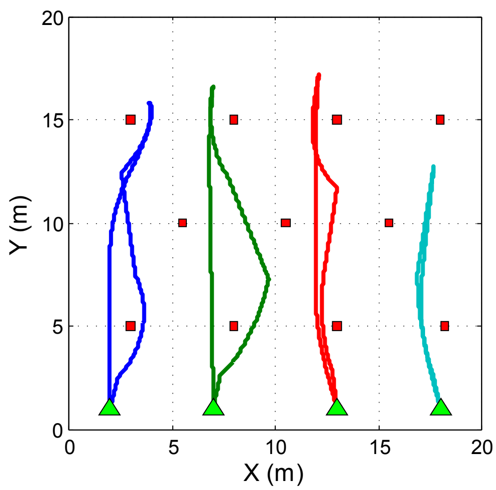
(b) Voronoi sampling regions in the $20 \times 20m^2$ map.



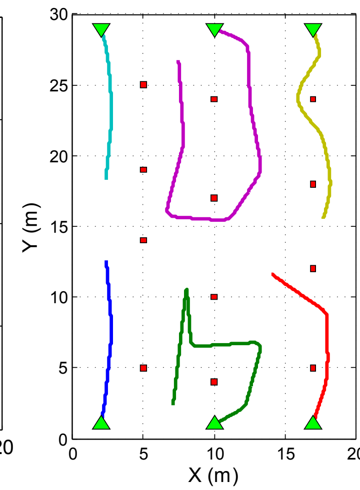
(c) Voronoi sampling regions in the $20 \times 30m^2$ map.



(d) Trajectories performed by the 2 UAV agents in the $9 \times 10m^2$ map.



(e) Trajectories performed by the 4 UAV agents in the $20 \times 20m^2$ map.



(f) Trajectories performed by the 6 UAV agents in the $20 \times 30m^2$ map.

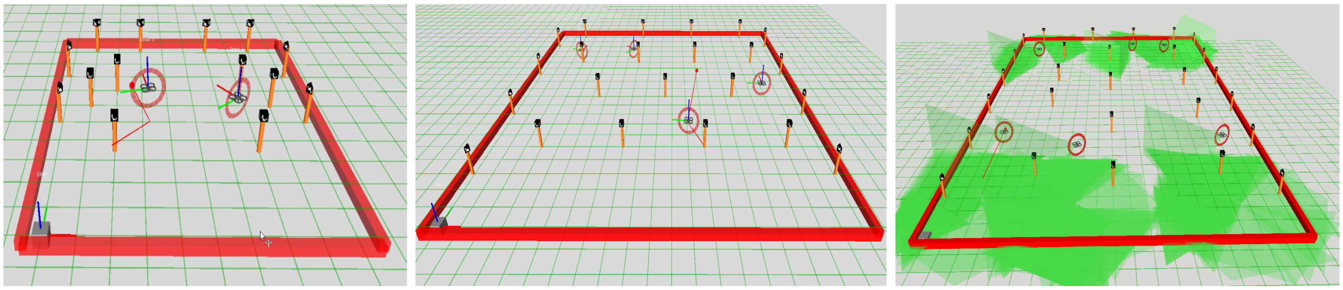
Fig. 4: Simulation results obtained in the different experiments conducted. Figs. (a), (b), (c), Voronoi regions generated by sampling the Mission Zone. “X” white marks correspond to the centroids of the clusters that will be used as mission points. Figs. (d), (e), (f), correspond to the trajectories generated for the different agents in the swarm while completing their respective missions. Green triangles depict the take-off point of the different UAVs in the swarm. Red squares represent the obstacles.

where the target is located at point (3, 15, 1.3). As it can be noticed in the figures, the trajectories are planned for visiting all the mission points avoiding the obstacles and searching for the target. In both experiments, the target was found by the UAV agent with the blue trajectory, and therefore trajectories of going back to the initial take-off point were commanded.

In Fig. 3b, it can be appreciated that the 2 agents in the swarm reached the final mission point almost at the same time. Thus, the trajectories of going back to the initial point depart almost from the same “y” coordinate in the Mission Zone.

Results obtained in simulated flights are shown in Figures 4a, 4d, for the $9 \times 10m^2$ map, and Figures 4b, 4e for the $20 \times 20m^2$ map. The trajectories performed by the UAVs in the swarm during the mission are shown in Figures 4d, 4e using different colors for each UAV. In Fig. 4d, it can be very well appreciated that the agent performing the blue trajectory locates the target in first place, sending the other agent in the swarm to the initial point. This agent stopped the mission that it was performing and went back to the initial point. Similar behavior can be noticed in Fig. 4e.

- Explore. In this type of mission the only input to the GMP is the high-level command of *EXPLORE*. The aim of this mission is to explore all the regions in the provided Mission Zone. For this purpose, the normal behavior of the agents of the swarm will be similar as explained in the *Find Target* mission, that is, move to the assigned mission point, turn in yaw to complete a 360° turn in each mission point, and perform a landing maneuver when the last mission point is reached.



(a) "Find Target" mission simulated in the $9 \times 10m^2$ map.

(b) "Find Target" mission simulated in the $20 \times 20m^2$ map.

(c) "Explore" mission simulated in the $20 \times 30m^2$ map.

Fig. 5: Pictures obtained from the experiments conducted in simulated flights using ROS Rviz. Fig. 5a, Simulation environment for the "Find Target" mission in a $9 \times 10m^2$ map, using 2 UAVs. Fig. 5b, Simulation environment for the "Find Target" mission in a $20 \times 20m^2$ map, using 4 UAVs. Fig. 5c, Simulation environment for the "Explore" mission in a $20 \times 30m^2$ map, using 6 UAVs. Here, green color represents the explored area that has been covered by the swarm.

The mission will be finished when all the agents in the swarm have reached the corresponding mission points assigned by the GMP in the respective missions.

Results of the experiments conducted for the *Explore* mission are shown in Figures 4c, 4f. The trajectories performed by the UAVs in the swarm during the mission are shown in Figure 4f using different colors for each UAV. As can be seen in Figure 4f, each UAV agent in the swarm has been assigned to a certain number of mission points, and performs the corresponding trajectories for the accomplishment of the mission.

The simulation environment used for the experiments presented in this section, is shown in Figure 5. In this Figure, the different trajectories planned by the agents in the swarm, while performing an obstacle avoidance maneuver, are shown as red lines. In Figure 5c, the area explored by the swarm is simulated (green color), taking into account the camera parameters of the ArDrone 2.0 and the size of the ArUco visual markers.

We refer the reader to a video[2], in which the capability of the proposed approach is demonstrated, showing the results obtained in the environments explained in section IV-B.

V. CONCLUSIONS AND FUTURE WORK

Collaborative Mission Planning using a swarm of UAVs is a very challenging task. To achieve the objective of designing an adaptable and dynamic mission planning architecture that can operate in real time with the desired scalability, research efforts must aim towards developing flexible and dynamic architectures that can be easily adapted for performing heterogeneous missions, and can be integrated with other modules in the architecture.

The current paper is an effort in this direction, with emphasis on developing a multi-UAV mission planning architecture that can provide a robust and fully operative framework for performing different high-level missions, such as Find Target or Explore, in a fully autonomous way.

[2] A video demonstration of the reported results has been made available at: https://youtu.be/2EHbb3y3UO8

A dynamic and scalable mission planning architecture for real flights applications, based on a Global Mission Planner (GMP) and an Agent Mission Planner (AMP), has been designed in this paper for addressing the problem of having a global intelligence for UAV swarm coordination. The GMP is able to translate a high-level global mission (e.g. Find Target) into several low level missions and tasks, and distribute them between the different UAV agents that conform the swarm, through the AMP, which is able to schedule and monitor in real time the tasks that have to be performed by each agent in the swarm.

In this work, several real flights have been conducted in an indoor scenario of $9 \times 10m^2$, for performing a high-level mission of "Find Target", using a swarm of 2 UAVs. With the aim of testing the scalability and flexibility of the proposed architecture, simulated flights have been conducted for performing the high-level mission of "Find Target" using a swarm of 4 UAVs in a scenario of $20 \times 20m^2$, and a mission of Exploration using a swarm of 6 UAVs in an scenario of $20 \times 30m^2$. Results obtained during simulated flights demonstrate the scalability and flexibility of the proposed mission planning architecture for a variable number of UAVs and different kind of scenarios, with variable number of obstacles.

Immediate future work is focused on the addition of several functionalities related with the global management of the behaviors of the swarm in case of failures, such as the dis-connectivity of the agents of the swarm from the ground station, failures in Motion Controllers, etc.

Another scope of research is lined towards exploring several optimization methods considering different directions, such as optimizing the trajectories that the agents in the swarm have to perform, optimizing the behavior of the swarm by means of the time required for performing the global mission, and optimization approaches based on the level of autonomy of the agents in the swarm. The development of the future work proposed above would lead in a better optimized and more robust architecture.

References

[1] P. Martin, O. Payton, J. Fardoulis, D. Richards, and T. Scott, "Yamashiki, y.(2016). low altitude unmanned aerial vehicle for characterising remediation effectiveness following the fdnpp accident. journal of environmental radioactivity, 153 (1), 58-63." *Journal of Environmental Radioactivity*, vol. 153, no. 1, pp. 58–63, 2015.

[2] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, *et al.*, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *Journal of Field Robotics*, vol. 29, no. 5, pp. 832–841, 2012.

[3] S. M. Adams and C. J. Friedland, *A survey of unmanned aerial vehicle (UAV) usage for imagery collection in disaster research and management*. publisher not identified, 2011.

[4] P. Gaudiano, E. Bonabeau, and B. Shargel, "Evolving behaviors for a swarm of unmanned air vehicles," in *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*. IEEE, 2005, pp. 317–324.

[5] G. B. Lamont, J. N. Slear, and K. Melendez, "Uav swarm mission planning and routing using multi-objective evolutionary algorithms," in *Computational Intelligence in Multicriteria Decision Making, IEEE Symposium on*. IEEE, 2007, pp. 10–20.

[6] P. Dasgupta, "A multiagent swarming system for distributed automatic target recognition using unmanned aerial vehicles," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 38, no. 3, pp. 549–563, 2008.

[7] J. D. Boškovic, N. Knoebel, N. Moshtagh, J. Amin, and G. L. Larson, "Collaborative mission planning & autonomous control technology (compact) system employing swarms of uavs," in *AIAA Guidance, Navigation, and Control Conference, Chicago*, 2009.

[8] Y. Wei, G. R. Madey, and M. B. Blake, "Agent-based simulation for uav swarm mission planning and execution," in *Proceedings of the Agent-Directed Simulation Symposium*. Society for Computer Simulation International, 2013, p. 2.

[9] Y. Wei, M. B. Blake, and G. R. Madey, "An operation-time simulation framework for uav swarm configuration and mission planning," *Procedia Computer Science*, vol. 18, pp. 1949–1958, 2013.

[10] J. L. Sanchez-Lopez, J. Pestana, P. de la Puente, R. Suarez-Fernandez, and P. Campoy, "A system for the design and development of vision-based multi-robot quadrotor swarms," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*. IEEE, 2014, pp. 640–648.

[11] J. L. Sanchez-Lopez, J. Pestana, P. de la Puente, and P. Campoy, "A reliable open-source system architecture for the fast designing and prototyping of autonomous multi-uav systems: Simulation and experimentation," *Journal of Intelligent & Robotic Systems*, pp. 1–19, 2015.

[12] S. Waharte and N. Trigoni, "Supporting search and rescue operations with uavs," in *Emerging Security Technologies (EST), 2010 International Conference on*. IEEE, 2010, pp. 142–147.

[13] P. Vincent and I. Rubin, "A framework and analysis for cooperative search using uav swarms," in *Proceedings of the 2004 ACM symposium on Applied computing*. ACM, 2004, pp. 79–86.

[14] H. V. D. Parunak, S. Brueckner, and J. J. Odell, "Swarming coordination of multiple uav's for collaborative sensing," in *Proceedings of Second AIAA" Unmanned Unlimited" Systems, Technologies, and Operations Conference*, 2003.

[15] R. R. McCune and G. R. Madey, "Agent-based simulation of cooperative hunting with uavs," in *Proceedings of the Agent-Directed Simulation Symposium*. Society for Computer Simulation International, 2013, p. 8.

[16] J. Gancet, G. Hattenberger, R. Alami, and S. Lacroix, "Task planning and control for a multi-uav system: architecture and algorithms," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 1017–1022.

[17] J. Sanchez-Lopez, R. Suarez-Fernandez, H. Bavle, C. Sampedro, M. Molina, J. Pestana, and P. Campoy, "Aerostack: An architecture and open-source software framework for aerial robotics," in *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*, June 2016.

[18] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[19] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.